Pratique AMMS : Estimation de la structure de la population à partir des données de fréquence des allèles

Izzy Routledge, Bob Verity, Nick Brazeau

August 05, 2022

Contents

Dépendances pour la pratique

Veuillez copier et coller le morceau de code ci-dessous dans son intégralité sur votre console pour télécharger les bibliothèques de packages R nécessaires à cette pratique. Si vous rencontrez des difficultés pour installer l'un des packages R, veuillez demander à un instructeur un lecteur flash préchargé.

```
deps <- c("tidyverse", "genescaperlite", "MIPanalyzer","rmaverick")
deps <- !sapply(deps, function(x){x %in% installed.packages()[,1]})
if (any(deps)) {
  if (deps["tidyverse"]) {
    install.packages("tidyverse")
  }
  if (deps["genescaper"]) {
    devtools::install_github("nickbrazeau/genescaperlite")
  }
  if (deps["MIPanalyzer"]) {
    devtools::install_github("mrc-ide/MIPanalyzer", ref = "v1.0.0")
  }
  if (deps["rmaverick"]) {
    devtools::install_github("bobverity/rmaverick", ref = "v1.1.0")
  }
}
```

Chargez maintenant toutes ces bibliothèques dans cette session en utilisant le morceau de code ci-dessous. Veuillez le copier-coller dans son intégralité.

```
library(tidyverse)
library(genescaperlite)
library(MIPanalyzer)
library(rmaverick)
```

Enfin, sourcez les fonctions supplémentaires nécessaires en copiant-collant cette fonction:

```
source("source_functions/pop_structure_utils.R")
```

Introduction à la structure de la population et aux mesures de la fréquence des allèles

L'une des observations les plus fondamentales que nous pouvons faire à partir des données génétiques est la façon dont les fréquences des différents allèles varient dans l'espace et dans le temps.

Pour les locus où les mutations confèrent une résistance aux médicaments, les fréquences alléliques nous disent quelque chose d'important sur le plan clinique : si un allèle est à des fréquences élevées, nous pourrions envisager de réaliser une étude d'efficacité thérapeutique (TES) et éventuellement de changer de médicament de première intention.

Cependant, même pour les loci qui n'ont pas d'impact direct sur le phénotype, et sont donc sélectivement neutres, les fréquences alléliques peuvent nous donner des informations importantes sur ce qui se passe dans la population de parasites.

Ces locus seront influencés par des facteurs tels que la prévalence, la taille des populations humaines et de moustiques et les taux de migration entre les différentes sous-populations. Si nous pouvons interpréter correctement ces signaux, nous pouvons utiliser des fréquences d'allèles neutres pour fournir des informations sur la structure et la connectivité de la population qui peuvent être pertinentes à des fins de contrôle.

Définition: *Un locus est une position fixe sur un chromosome où se trouve un marqueur génétique particulier.*

Définition: *Un allèle est une variante particulière d'un gène ou d'un locus génétique.*

Définition: *Un allèle neutre est un allèle qui n'affecte pas positivement ou négativement la forme physique d'un organisme.*

Aperçu des données

Nous travaillerons avec des données simulées et réelles dans cette pratique. Les données simulées nous permettront d'explorer comment les fréquences alléliques changent au fil du temps en fonction de différents paramètres sous-jacents - ce que nous ne pouvons presque jamais faire à partir de données désordonnées du monde réel. Cela nous permettra de construire une intuition sur ce que nous nous attendrions à voir dans les données réelles. Dans la seconde moitié de la pratique, nous utiliserons des ensembles de données du monde réel, y compris un grand ensemble de données de sonde d'inversion moléculaire (MIP) provenant de la RDC et des pays environnants, précédemment analysé par Verity et al. 2020, et un plus petit ensemble de données de codes-barres génétiques provenant du Sénégal, précédemment analysé par Bei et al., 2018.

Objectifs pratiques

À la fin de cet exercice pratique, vous devriez être en mesure de:

- Décrire comment la taille de la population, les taux de mutation et de migration affectent les modèles que nous nous attendons à voir dans les données de fréquence des allèles neutres
- Estimez des métriques telles que F_{ST} et expliquez ce qu'elles peuvent nous dire sur la différenciation de la population
- Utiliser des techniques d'ordination comme l'ACP pour explorer les modèles de structure de la population

- Utilisez des techniques basées sur des modèles comme *rmaverick* pour détecter la structure de la population de manière probabiliste
- Apprécier comment les méthodes de fréquence allélique peuvent identifier la connectivité sur de grandes échelles spatiales et de longues échelles temporelles
- Expliquer comment les fréquences alléliques peuvent donner des informations utiles et décrire également leurs limites à des fins de lutte contre le paludisme

---

Génétique des populations par simulation

Dans cette section, nous allons simuler différentes populations avec différentes fréquences d'allèles en utilisant la fonction R : `sim_freqs()`. Notez que cette fonction n'est pas présente dans la base R, mais a été chargée à partir de `source_functions/pop_structure_utils.R` qui à son tour fait référence au package `genescaper` (en développement). Cette fonction simule les données d'un modèle Wright-Fisher, qui est un modèle génétique de population générale et non spécifique au paludisme. Il ne parvient donc pas à capter certaines dynamiques du paludisme, comme la surinfection, mais il a l'avantage d'être très simple et facile à interpréter.

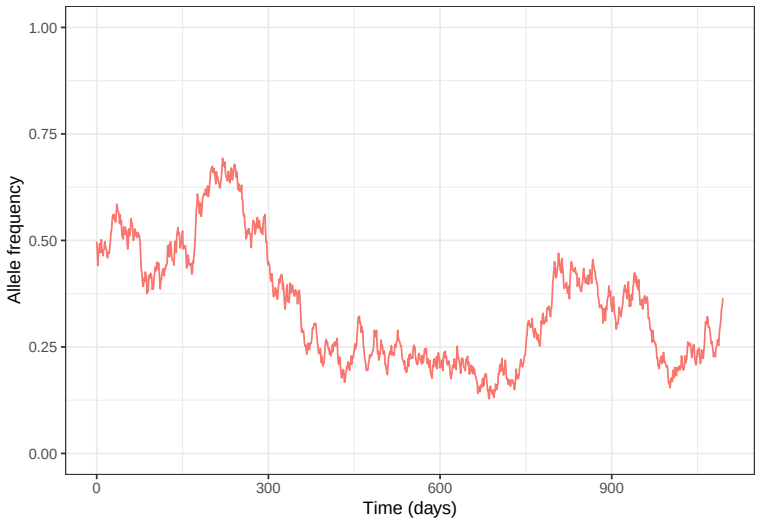
Sous-population unique

Considérons d'abord un seul "dème", ou sous-population. Nous pouvons considérer cela comme une petite population d'hôtes infectés, les parasites étant librement transmis d'un individu à un autre (c'est-à-dire sans structure de population humaine ou de moustiques).

La fréquence des allèles change dans un seul dème

Nous allons d'abord simuler les fréquences alléliques d'une population de 1000 individus infectés:

```
sim1 <- sim_freqs(N = 1000, mut_rate = 0)
```



Q1. Essayez d'exécuter cette fonction plusieurs fois. Que remarquez-vous au sujet de la fréquence des allèles au fil du temps ? Que se passe-t-il s'ils atteignent exactement 0 ou 1, et pourquoi?

Click For Answer

A1. Les fréquences alléliques fluctuent de manière aléatoire dans le temps. Ceci est le résultat de notre hypothèse de taille de population finie et d'accouplement aléatoire, ce qui signifie que certaines infections seront plus "réussies" que d'autres, transmettant plus d'infections secondaires. Les résultats sont différents à chaque fois que nous exécutons la fonction en raison de la nature aléatoire du processus. Parfois, les

fréquences alléliques peuvent atteindre exactement 0 ou 1 ("perte" ou "fixation"), auquel cas elles *restent à ce niveau*. En effet, nous n'avons supposé aucun taux de mutation ci-dessus, ce qui signifie qu'une fois qu'une population est fixée pour un seul allèle, il n'y a aucun moyen de faire revenir une nouvelle variation.

Q2. Maintenant, répétez avec une taille de population plus petite (qui représente le nombre d'hôtes *infectés* dans notre cas) ; par exemple, `N=500`, `N=100` et `N=50`. Que remarquez-vous à propos de la force de la dérive?

Click For Answer

A2. La force de la dérive génétique (c'est-à-dire l'ampleur des fluctuations des fréquences alléliques) est maximale lorsque la taille de la population est petite. Nous avons plus de chances d'atteindre la fixation ou la perte plus rapidement lorsque la taille de la population est petite. Du point de vue du paludisme, cela signifie que nous sommes plus susceptibles de voir un seul allèle dominer une population par pur hasard lorsque le nombre d'hôtes infectés est faible (c'est-à-dire une faible prévalence ou une petite taille de population).

Définition: *La dérive génétique est le processus de fluctuations aléatoires des fréquences alléliques dans une population finie.*

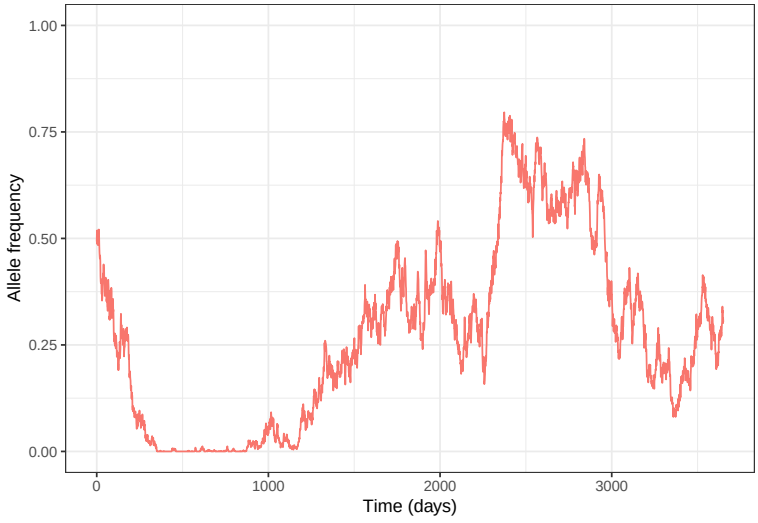
Jusqu'à présent, nous n'avons supposé aucune mutation, ce qui signifie que la seule force affectant toutes les fréquences est la dérive. Ensuite, nous explorerons comment nos résultats de simulation changent lorsque nous ajoutons une mutation. Les taux de mutation que nous considérerons ici sont supérieurs de plusieurs ordres de grandeur aux taux de mutation SNP typiques chez *P. falciparum*, qui sont autour de `1e-9`. Ce n'est que pour des raisons de commodité - nous pourrions obtenir les mêmes résultats avec un taux de mutation plus faible, mais nous aurions besoin d'une taille de population plus grande pour voir les effets qui seraient lents à simuler.

Q3. Répétez la simulation ci-dessus, cette fois sur 10 ans. Commencez par supposer une taille de population de 1000 individus infectés et un taux de mutation de "1e-4". Les fréquences finissent-elles toujours par être fixées/perdues ? Que se passe-t-il si vous augmentez le taux de mutation ? Que se passe-t-il si vous diminuez la taille de la population ?

Click For Answer

A3. Les allèles ont tendance à atteindre 0 et 1 sur une si longue période de temps, mais ils reviennent ensuite à des fréquences intermédiaires. Cela est dû à une mutation réintroduisant une petite quantité de variation, que la dérive peut ensuite ramener vers les hautes fréquences. Des taux de mutation plus élevés signifient que les fréquences alléliques ont tendance à rester à des valeurs intermédiaires plus longtemps. Des tailles de population plus petites signifient que la dérive est plus forte (comme ci-dessus), de sorte que les fréquences alléliques ont tendance à rester proches de 0 ou 1.

```
sim_freqs(N = 1000, t_out = seq(0, 365 * 10), mut_rate = 1e-4)
```



La distribution de fréquence des allèles

Nous allons maintenant simuler un grand nombre de lieux, mais la sortie ne prendra en compte qu'un seul point de temps : 50 ans à compter du début de notre simulation.

```
# simulate allele frequencies and look at the data
sim1 <- sim_freqs(N = 1e3, t_out = 365 * 50, mut_rate = 1e-4, loci = 1e3, plot_on = FALSE)
head(sim1$data)
```

```
##    time deme locus  freq
## 1 18250    1     1 0.999
## 2 18250    1     2 0.116
## 3 18250    1     3 0.999
## 4 18250    1     4 0.552
## 5 18250    1     5 0.011
## 6 18250    1     6 0.035
```

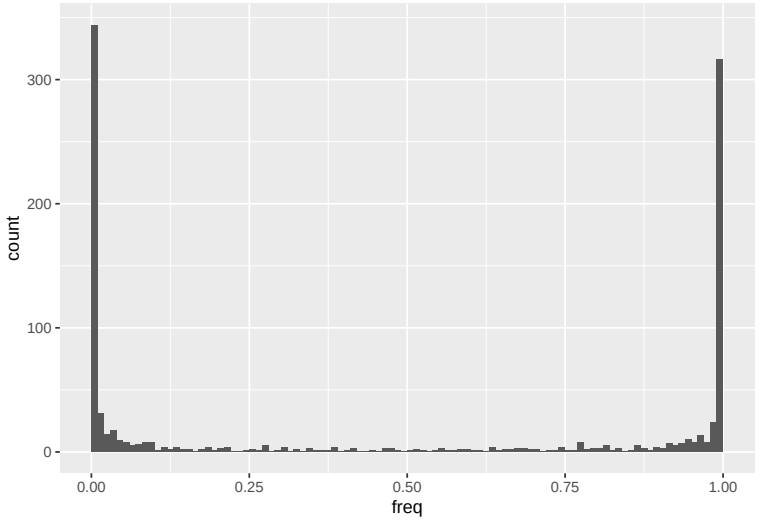
Nous pouvons voir que nos différents locus atteignent différentes fréquences d'allèles en raison du hasard. Il peut être utile d'examiner la **distribution de fréquence des allèles**, parfois aussi appelée le **spectre de fréquence des allèles**, pour voir à quelle fréquence nous voyons chaque fréquence.

Q4. À l'aide des données simulées situées dans la trame de données `sim1$data`, créez un histogramme des fréquences d'allèles. Essayez d'utiliser ggplot avec la fonction `geom_histogram()`. Vous devrez peut-être modifier l'argument `breaks` de cette fonction pour voir clairement la distribution de fréquence.

Click For Answer

A4. Un exemple de code est présenté ci-dessous :

```
ggplot(sim1$data) +
  geom_histogram(aes(x = freq), breaks = seq(0, 1, 0.01))
```

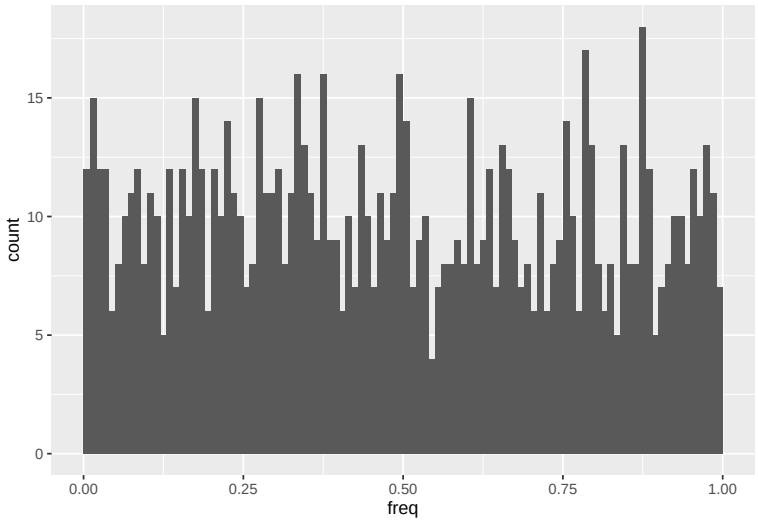


Q5. Qu'advient-il de la distribution de fréquence des allèles si vous augmentez le taux de mutation dans votre simulation ? Que se passe-t-il si vous augmentez la taille de la population ?

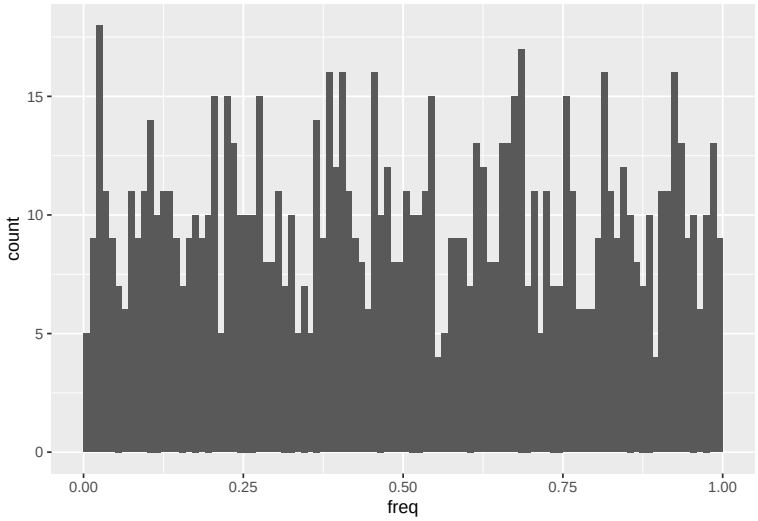
Click For Answer

A5. L'exemple de code ci-dessous montre des simulations avec un taux de mutation plus élevé ou une taille de population plus importante qu'auparavant. Dans les deux cas, nous voyons que la distribution de la fréquence des allèles est plus concentrée aux valeurs centrales, plutôt que d'être poussée contre les murs à 0 et 1. C'est parce que la mutation est forte *par rapport à la dérive*, ce qui signifie que les fréquences alléliques intermédiaires peuvent être maintenues.

```r
# simulate and plot large mutation rate
sim2 <- sim_freqs(N = 1e3, t_out = 365 * 50, mut_rate = 1e-3, loci = 1e3, plot_on = FALSE)
ggplot(sim2$data) +
  geom_histogram(aes(x = freq), breaks = seq(0, 1, 0.01))
```




```r
# simulate and plot large population size
sim3 <- sim_freqs(N = 1e4, t_out = 365 * 50, mut_rate = 1e-4, loci = 1e3, plot_on = FALSE)
ggplot(sim3$data) +
  geom_histogram(aes(x = freq), breaks = seq(0, 1, 0.01))
```



Les distributions de fréquence des allèles SNP dans le monde réel ont tendance à ressembler davantage à notre première simulation, avec la plupart des valeurs proches de 0 ou 1. En effet, la taille effective de la population dans *P. falciparum* est très grand et le taux de mutation est également assez faible. Rappelez-vous que cette distribution correspond à ce que nous attendons uniquement des allèles * neutres * - des choses comme l'équilibrage de la sélection peuvent créer des modèles plus compliqués, tendant à maintenir les fréquences à des valeurs intermédiaires pendant de longues périodes.

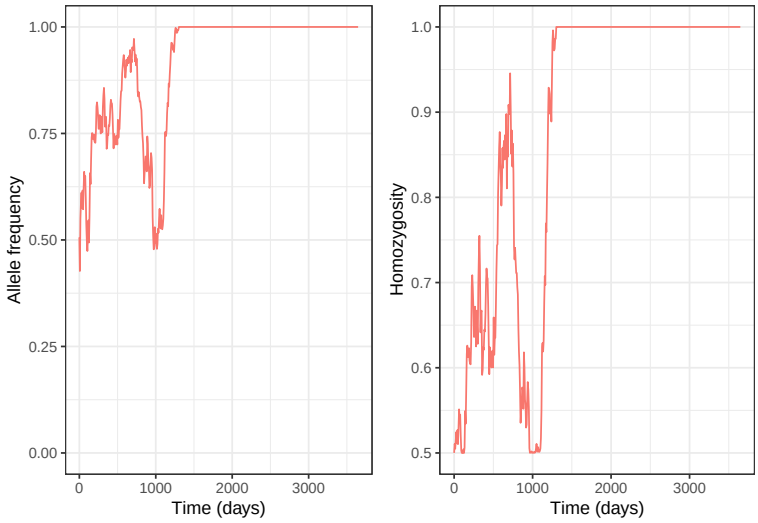
L'accumulation d'homozygotie

Nous allons maintenant analyser comment l'homozygotie évolue dans le temps en raison de la dérive génétique.

Définition: *L'homozygotie est l'existence d'allèles identiques à un locus donné. Il peut également être utilisé pour décrire la fréquence ou la probabilité de voir des allèles identiques.*

Q6. Exécutez le code ci-dessous plusieurs fois pour répéter la simulation de dérive sans mutation, mais maintenant également en traçant l'homozygotie. Que remarquez-vous sur la façon dont l'homozygotie change avec le temps?

```
sim_freqs(N = 1000, t_out = seq(0, 365 * 10, 7), plot_homo = TRUE)
```



Click For Answer

A6. L'homozygotie a tendance à augmenter jusqu'à 1 avec le temps. Ceci est vrai quel que soit le sens de dérive de la fréquence allélique (c'est-à-dire vers 0 ou vers 1). L'homozygotie atteint sa valeur finale de 1 au moment où les fréquences alléliques se fixent à 0 ou 1.

Q7. Exécutez le code ci-dessus, mais considérez maintenant ce qui se passe lorsque vous ajoutez une mutation ? Essayez des taux de mutation de `1e-3` et `1e-4`. Que devient le niveau d'équilibre d'homozygotie ?

Click For Answer

A7. Les fréquences alléliques ont tendance à être tirées vers des valeurs intermédiaires, ce qui signifie que l'homozygotie est éloignée de 1. Même si l'homozygotie finit par atteindre 1, la mutation peut signifier que la variation est réintroduite, de sorte que l'homozygotie chute à nouveau. Le niveau d'équilibre d'homozygotie n'est donc pas exactement 1 lorsqu'il y a une mutation présente.

```
# explore homozygosity trends when there is some mutation
sim_freqs(N = 1000, t_out = seq(0, 365 * 10, 7), mut_rate = 1e-3, plot_homo = TRUE)
sim_freqs(N = 1000, t_out = seq(0, 365 * 10, 7), mut_rate = 1e-4, plot_homo = TRUE)
```

Sous-populations multiples

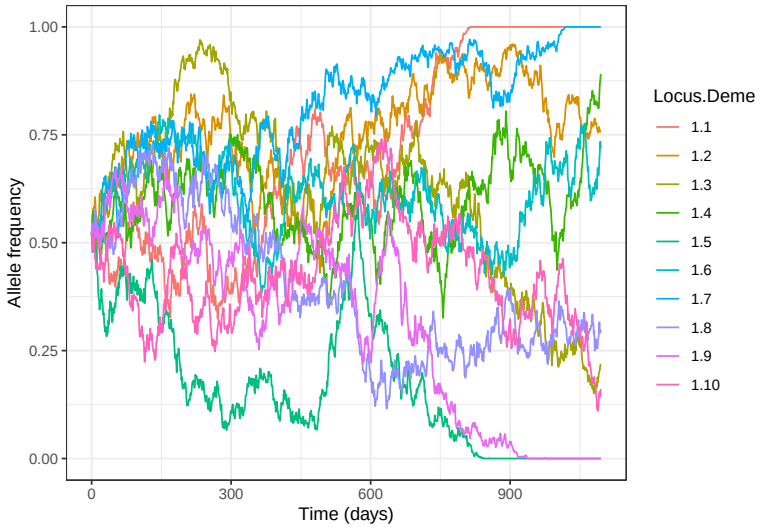
Nous avons vu comment les fréquences alléliques changent avec le temps en raison de la mutation et de la dérive dans une seule sous-population. Dans cette section, nous explorerons comment ces schémas varient entre les sous-populations et comment la migration influence ces schémas.

Sous-populations parfaitement isolées

Tout d'abord, nous allons explorer un scénario où nous avons 10 dèmes parfaitement indépendants. Cela pourrait représenter dix villages où il n'y a pas de mouvement de personnes (ou de parasites) entre ces villages.

Exécutez le code suivant, qui simule la dérive dans 10 dèmes :

```
sim1 <- sim_freqs(N = 1000, demes = 10, mig_rate = 0)
```



Vous devriez voir que les fréquences alléliques ont tendance à diverger entre les dèmes. C'est le même comportement que nos simulations de sous-population unique ci-dessus, mais maintenant reproduit 10 fois. Les fréquences des allèles ont une chance égale d'augmenter ou de diminuer un jour donné, ce qui signifie que certains dèmes augmenteront et d'autres diminueront et donc, dans l'ensemble, nous constatons une divergence.

Q8. À quoi vous attendriez-vous si la population était plus nombreuse ? Qu'en est-il si vous introduisez la mutation ?

Click For Answer

A8. Avec une plus grande taille de population, la dérive est plus faible, ce qui signifie que les populations divergent mais cela prend plus de temps. Avec certaines mutations, nous constatons que les fréquences alléliques sont ramenées vers des valeurs intermédiaires, ce qui signifie qu'il y a une limite à la divergence (nous ne nous retrouvons pas toujours avec des allèles fixes ou perdus dans chaque dème).

```
sim1 <- sim_freqs(N = 1e4, demes = 10, mig_rate = 0)
sim1 <- sim_freqs(N = 1e3, demes = 10, mig_rate = 0, mut_rate = 1e-4)
```

Calcul de la différenciation des populations Ensuite, nous explorerons les métriques utilisées pour mesurer la différenciation des populations. Nous allons nous concentrer sur deux métriques différentes : l'*indice de fixation*, ou F_{ST} et *Jost's D*.

Des valeurs élevées de F_{ST} et du D de Jost suggèrent toutes deux une plus grande différenciation génétique entre les sous-populations. Cependant, les deux mesures ont des approches différentes et des définitions différentes de ce que nous entendons exactement par "différenciation" génétique.

Fst Nous examinerons d'abord F_{ST}, qui mesure à quel point nos populations sont proches de la fixation. Au fur et à mesure que nos populations se différencieront, dans le sens où elles accumuleront une ascendance locale, elles se rapprocheront également de la fixation (en supposant zéro mutation). Par conséquent, il est courant d'interpréter F_{ST} comme une mesure de différenciation ainsi qu'une mesure de progrès vers la fixation.

Ci-dessous, nous avons fourni une version simple de F_{ST} qui s'applique uniquement aux loci bialléliques et peut être calculée manuellement:

$$F_{ST} = \frac{\sigma^2_{\text{subpop}}}{\sigma^2_{\text{total}}} = \frac{\sigma^2_{\text{subpop}}}{\bar{p}(1-\bar{p})}$$

Cette équation indique que F_{ST} est une mesure de la variance des fréquences alléliques parmi les sous-populations divisée par la variation totale de la fréquence allélique dans la population.

```
# simulate allele frequencies as above
sim1 <- sim_freqs(N = 1e3, demes = 10, t_out = seq(0, 365*10, 7), mig_rate = 0, plot_on = FALSE)

# calculate Fst at each timepoint
sim_fst <- sim1$data %>%
  select(-locus) %>%
  group_by(time) %>%
  summarise(p_mean = mean(freq),
            p_var = var(freq),
            Fst = p_var / (p_mean * (1 - p_mean)))

# look at first few results
head(sim_fst)
```

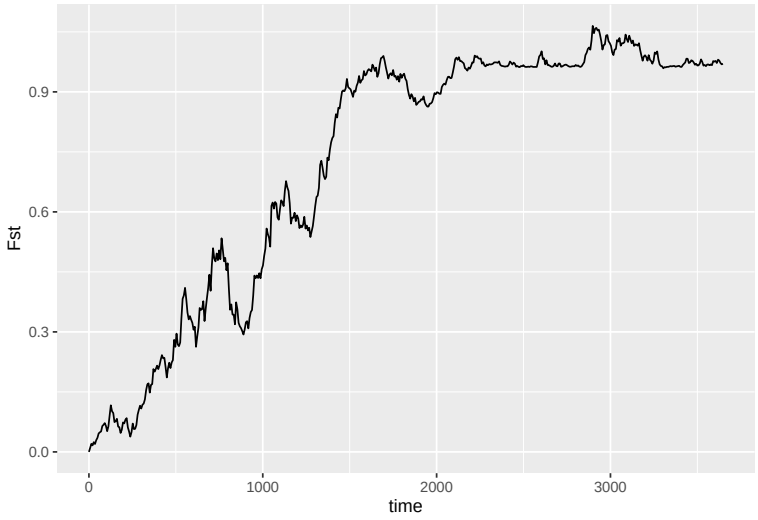
```
## # A tibble: 6 x 4
##    time p_mean   p_var     Fst
##   <dbl>  <dbl>   <dbl>   <dbl>
## 1     0  0.454 0       0
## 2     7  0.452 0.00245 0.00989
## 3    14  0.431 0.00497 0.0203
## 4    21  0.454 0.00404 0.0163
## 5    28  0.466 0.00608 0.0244
## 6    35  0.452 0.00487 0.0197
```

Q9. Pouvez-vous utiliser la fonction ggplot2 `geom_line()` pour tracer F_{ST} dans le temps (axe x : temps, axe y : Fst). Que remarquez-vous au sujet de la différenciation ?

Click For Answer

A9. La différenciation (mesurée par Fst) augmente avec le temps. Cela est dû à la divergence des fréquences alléliques, ce qui fait que la variance entre les dèmes représente une proportion croissante de la variance totale.

```
ggplot(sim_fst) +
  geom_line(aes(x = time, y = Fst)) +
  expand_limits(y = c(0, 1))
```

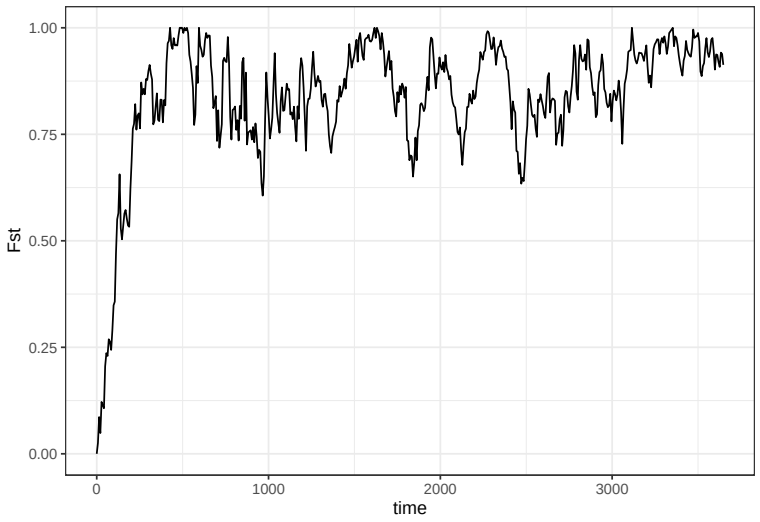


Q10. Répétez maintenant le processus ci-dessus (simulation et tracé) avec une taille de population de `N = 1e2` et un taux de mutation de `mut_rate = 1e-3`. Quels changements remarquez-vous ?

Click For Answer

A10. F_{ST} augmente avec le temps, mais ne se stabilise pas à 1. Au lieu de cela, il rebondit à une valeur d'équilibre inférieure à 1.

```
# simulate with a high mutation rate
sim1 <- sim_freqs(N = 1e2, demes = 10, mig_rate = 0, mut_rate = 1e-3,
                  t_out = seq(0, 365*10, 7), plot_on = FALSE)

# calculate Fst and plot results
sim1$data %>%
  select(-locus) %>%
  group_by(time) %>%
  summarise(p_mean = mean(freq),
            p_var = popvar(freq),
            Fst = p_var / (p_mean * (1 - p_mean))) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = time, y = Fst)) +
  expand_limits(y = c(0, 1))
```



Le niveau d'équilibre de F_{ST} entre la dérive et la mutation est connu pour être $\frac{1}{(1+2N\mu)}$ pour les organismes haploïdes, où N est la taille de la population, et mu est le taux de mutation.

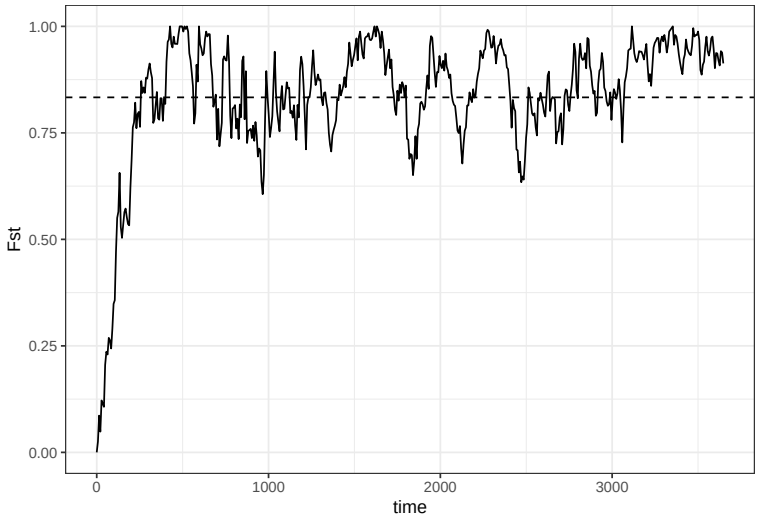
Q11. Ajoutez ce niveau dans le tracé ci-dessus à l'aide de la fonction `geom_hline()`. Comment ce résultat théorique se compare-t-il à ce que vous voyez dans l'intrigue ?

Click For Answer

A11. Le résultat théorique correspond de très près à ce que nous observons dans le graphique.

```
# plot results as above, adding line for theoretical expectation
sim1$data %>%
  select(-locus) %>%
  group_by(time) %>%
  summarise(p_mean = mean(freq),
            p_var = popvar(freq),
            Fst = p_var / (p_mean * (1 - p_mean))) %>%
  ggplot() + theme_bw() +
```

```
  geom_line(aes(x = time, y = Fst)) +
  expand_limits(y = c(0, 1)) +
  geom_hline(yintercept = 1 / (1 + 2*1e2*1e-3), linetype = "dashed")
```



Le fait que F_{ST} dépende du taux de mutation peut entraîner des problèmes. Si un marqueur a un taux de mutation élevé (par exemple des microsatellites), on s'attendrait à voir de faibles valeurs de F_{ST}, même si les populations évoluent séparément depuis longtemps, et en ce sens sont bien différenciées. Pour cette raison, nous préférons peut-être envisager d'autres mesures.

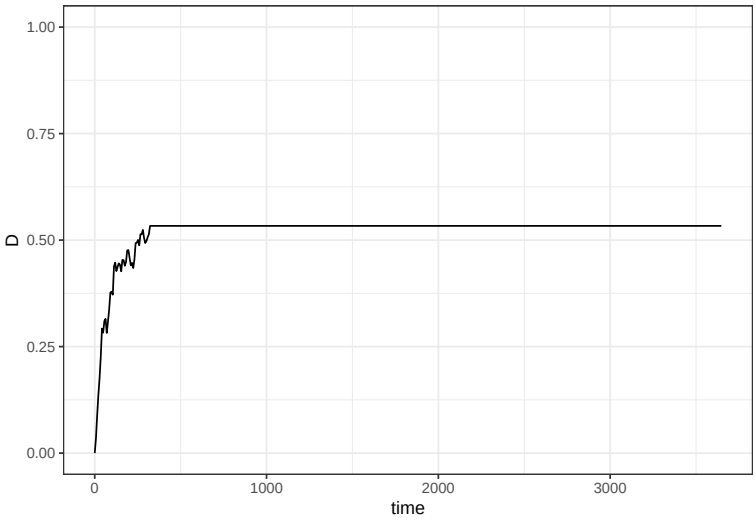
D de Jost Le D de Jost est une autre façon d'envisager la différenciation qui ne souffre pas des mêmes problèmes de mutation. Nous pouvons calculer le D de Jost, en utilisant l'équation suivante :

$$D = \frac{H_T - H_S}{1 - H_S} \frac{k}{k-1}$$

Le D de Jost mesure l'étendue du partage d'allèles entre les dèmes. Lorsque tous les allèles sont "privés" à un seul dème, ce qui signifie que le même allèle n'est jamais vu dans plus d'un dème, alors le D de Jost est égal à 1. Il s'agit d'une mesure de "différenciation" en termes de partition de la diversité génétique observée, et donc est une interprétation complètement différente de F_{ST}.

Nous pouvons utiliser le morceau de code ci-dessous pour explorer cette statistique :

```
# simulate with a high mutation rate
sim1 <- sim_freqs(N = 1e2, demes = 10, mig_rate = 0, mut_rate = 0,
                  t_out = seq(0, 365*10, 7), plot_on = FALSE)

# calculate Jost's D and plot results
sim1$data %>%
  select(-locus) %>%
  group_by(time) %>%
  summarise(Hs = 1 - mean(freq^2 + (1 - freq)^2),
            Ht = 1 - (mean(freq)^2 + mean(1 - freq)^2),
            k = max(deme),
            D = (Ht - Hs) / (1 - Hs) * k / (k - 1)) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = time, y = D)) +
  expand_limits(y = c(0, 1.0))
```



Le D de Jost ne souffre pas du même problème que F_{ST}, mais il souffre d'un problème différent - il suppose une mutation infinie des allèles. Lorsque nous avons un SNP biallélique, le D de Jost aura tendance à se stabiliser à une valeur inférieure à 1, et est également difficile à prédire. Ce serait une statistique beaucoup plus appropriée pour quelque chose comme un microsatellite, où la mutation des allèles infinis est plus raisonnable.

En conclusion, F_{ST} et D ont leurs limites. F_{ST} est bon lorsque les taux de mutation sont faibles, par exemple les SNP biallélliques. Le D de Jost convient aux sites très divers et à mutation rapide, tels que les microsatellites.

Sous-populations connectées

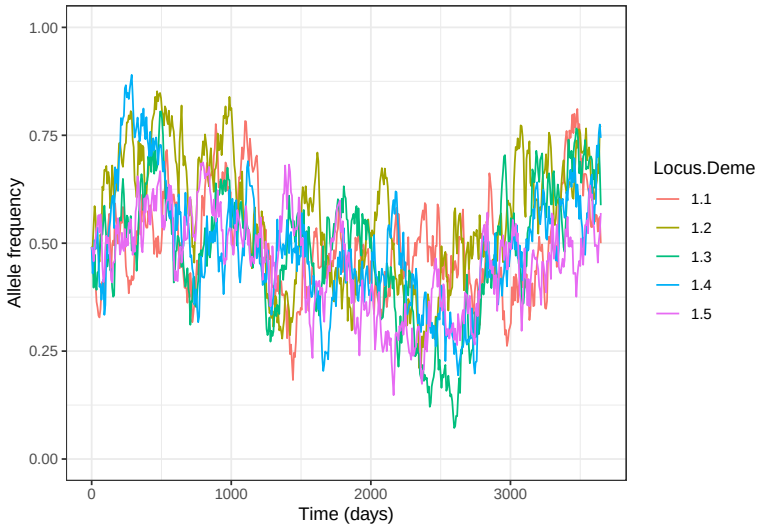
Dans cette section, nous explorerons ce qui se passe lorsque nous connectons nos dèmes par la migration. En d'autres termes, nous assouplissons notre hypothèse de populations indépendantes d'en haut.

Q12. Exécutez le simulateur maintenant avec 1000 individus, 5 dèmes, un taux de migration de 1 sur 100 et un taux de mutation de 1 sur 10 000. Que remarquez-vous sur les fréquences alléliques ? Essayez d'expérimenter avec des taux de migration plus grands et plus petits. Comment cela change-t-il les résultats ?

Click For Answer

A12. Avec la migration, les fréquences alléliques fluctuent toujours mais ont tendance à se déplacer ensemble. Plus le taux de migration est élevé, plus les résultats sont étroitement corrélés. Vous remarquerez peut-être également que les fluctuations sont plus faibles (la dérive est plus faible) lorsque la migration est élevée, car notre taille de population "effective" devient plus grande.

```
# simulation and plot with migration
sim_freqs(N = 1000, t_out = seq(0, 365*10, 7), demes = 5, mut_rate = 1e-4, mig_rate = 1e-2)
```



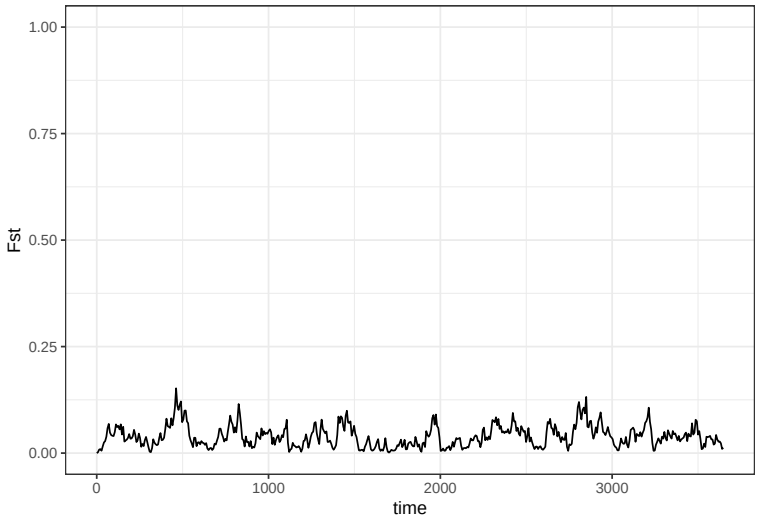
Fst avec migration Regardons ce qui arrive à F_{ST} lorsque nous incluons la migration.

Q13. Exécutez la simulation ci-dessus et utilisez le code que vous avez vu précédemment pour calculer et tracer Fst à partir des résultats. Qu'advient-il de F_{ST} lorsque vous augmentez/diminuez le taux de migration ? Est-ce que ça a du sens?

Click For Answer

A13. Avec la migration, les fréquences alléliques fluctuent toujours mais ont tendance à se déplacer ensemble. Plus le taux de migration est élevé, plus les résultats sont étroitement corrélés. Vous remarquerez peut-être également que les fluctuations sont plus faibles (la dérive est plus faible) lorsque la migration est élevée, car notre taille de population "effective" devient plus grande.

```
# define parameters
N <- 1e3
mu <- 1e-4
m <- 1e-2

# simulate data with migration and mutation
sim1 <- sim_freqs(N = N, t_out = seq(0, 365*10, 7), demes = 5, mut_rate = mu,
                  mig_rate = m, plot_on = FALSE)

# calculate and plot Fst
sim1$data %>%
  select(-locus) %>%
  group_by(time) %>%
  summarise(p_mean = mean(freq),
            p_var = popvar(freq),
            Fst = p_var / (p_mean * (1 - p_mean))) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = time, y = Fst)) +
  expand_limits(y = c(0, 1))
```



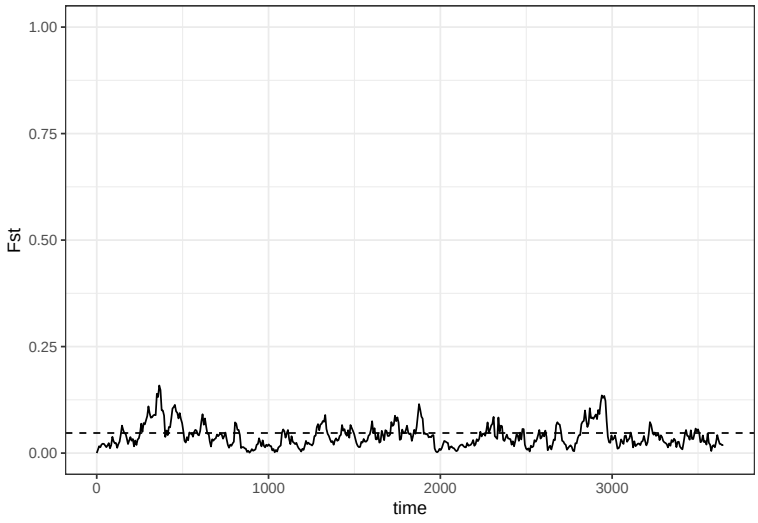
La théorie nous dit que le niveau d'équilibre de F_{ST} dans cette situation est donné par $\frac{1}{1+2N(m+\mu)}$. C'est une idée similaire à l'équilibre dont nous avons discuté ci-dessus, mais nous équilibrons maintenant les forces de dérive, de migration et de mutation pour atteindre un nouvel équilibre.

Q14. Votre simulation correspond-elle à cette valeur d'équilibre ? Notez que les taux de migration sont généralement beaucoup plus élevés que les taux de mutation, ce qui signifie que nous pouvons souvent simplifier l'équilibre ci-dessus en $\frac{1}{1+2Nm}$.

Click For Answer

A14. La simulation correspond très bien à la théorie. Lorsque les taux de migration sont élevés, les valeurs de Fst sont souvent très faibles.

```
# define parameters
N <- 1e3
mu <- 1e-4
m <- 1e-2

# calculate theoretical equilibrium value
Fst_eq <- 1 / (1 + 2*N*(m + mu))

# simulate data with migration and mutation
sim1 <- sim_freqs(N = N, t_out = seq(0, 365*10, 7), demes = 5, mut_rate = mu,
                  mig_rate = m, plot_on = FALSE)

# calculate and plot Fst, and overlay equilibrium
sim1$data %>%
  select(-locus) %>%
  group_by(time) %>%
  summarise(p_mean = mean(freq),
            p_var = popvar(freq),
            Fst = p_var / (p_mean * (1 - p_mean))) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = time, y = Fst)) +
  expand_limits(y = c(0, 1)) +
  geom_hline(yintercept = Fst_eq, linetype = "dashed")
```



En conclusion, à l'équilibre, F_{ST} atteint un niveau qui est un équilibre entre la force de la dérive génétique et la migration. La mutation joue un rôle, mais le taux de mutation est souvent beaucoup plus faible que le taux de migration, ce qui signifie que nous pouvons effectivement l'ignorer. Plus le taux de migration est élevé, plus la valeur de F_{ST} est faible. Cela signifie que nous pouvons utiliser F_{ST} pour nous dire quelque chose sur la connectivité des populations. Les petits F_{ST} par paires entre les dèmes tendent à indiquer des populations fortement connectées et vice versa.

Notez cependant que ces modèles prennent beaucoup de temps à se développer. La plupart des simulations ci-dessus sont exécutées sur de nombreuses années, et ceci pour une petite taille de population - pour une plus grande taille de population, cela peut prendre des décennies pour atteindre l'équilibre. Ainsi, les changements de fréquence des allèles et les métriques de différenciation peuvent nous dire quelque chose sur la connectivité sur une **échelle de temps évolutive**. Cela fournit un contexte de fond important, mais n'est probablement pas directement pertinent à des fins de contrôle.

Analyse des données de la RDC

Nous allons maintenant explorer les fréquences alléliques dans un jeu de données réel. Tout d'abord, nous allons charger un ensemble de données collectées à l'aide de sondes d'inversion moléculaire (MIP) qui sont une méthode de séquençage à haut débit. Cet ensemble de données se compose de 2537 échantillons collectés en 2013-2015 en RDC et dans les pays environnants. Ce jeu de données est une version légèrement simplifiée de celui utilisé dans l'article de Verity et al. 2020.

```
# load data
MIP_data <- readRDS("data/DRC_MIPs_biallelic_processed.rds")

# have a look at the data
names(MIP_data)
```

```
## [1] "coverage"       "counts"         "samples"        "loci"
## [5] "filter_history" "vcfmeta"
```

```
head(MIP_data$samples)
```

```
##                       ID Country Admin1_name Year Latitude Longitude Cluster
## 07GHR5002-AG-1 07GHR5002   Ghana        <NA> 2013 6.385908 -0.376496       1
## 07GHR5006-AG-1 07GHR5006   Ghana        <NA> 2013 6.385908 -0.376496       1
## 07GHR5018-AG-1 07GHR5018   Ghana        <NA> 2013 6.385908 -0.376496       1
```

```
## 07GHR5025-AG-1 07GHR5025   Ghana        <NA> 2013 6.385908 -0.376496       1
## 07GHR5028-AG-1 07GHR5028   Ghana        <NA> 2013 6.385908 -0.376496       1
## 07GHR5029-AG-1 07GHR5029   Ghana        <NA> 2013 6.385908 -0.376496       1
```

```
head(MIP_data$loci)
```

```
##    CHROM    POS REF ALT     NEUTRAL           GEO
## 9      1 100608   A   G     Neutral Non-geographic
## 19     1 138823   C   T     Neutral Non-geographic
## 31     1 139191   C   T Non-neutral     Geographic
## 35     1 140820   A   C Non-neutral     Geographic
## 36     1 155939   G   A     Neutral Non-geographic
## 63     1 182927   T   C     Neutral Non-geographic
```

Les ensembles de données MIP comme celui-ci peuvent être analysés à l'aide du package `MIPanalyzer`, qui contient diverses fonctions de filtrage, d'analyse et de visualisation. Par exemple, nous pouvons créer une copie de l'ensemble de données qui filtre tous les échantillons DRC à l'aide de la fonction `MIPanalyzer::filter_samples()`:

```
# filter samples
MIP_data_noDRC <- MIPanalyzer::filter_samples(MIP_data, MIP_data$samples$Country != "DRC")

MIP_data_noDRC
```

PCA des données de la RDC

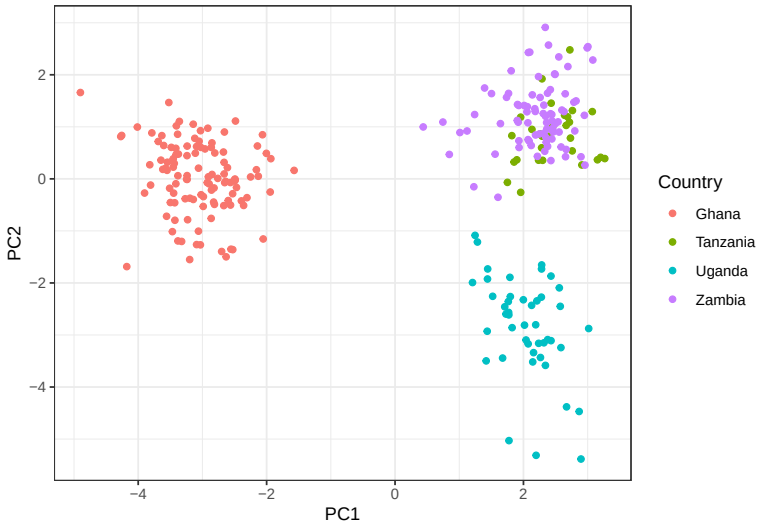
L'analyse en composantes principales, ou PCA, peut être utilisée pour réduire le nombre de dimensions ou de variables dans un ensemble de données. En tant que tel, il est souvent utilisé comme moyen de visualiser des données de grande dimension, telles que les fréquences alléliques dans cet exemple.

Tout d'abord, nous pouvons utiliser la fonction `get_wsaf()` pour calculer les fréquences alléliques intra-échantillon à chaque locus et pour chaque individu. Ensuite, nous pouvons utiliser la fonction `pca_wsaf()` pour calculer la sortie PCA à partir de ces fréquences :

```
# calculate within-sample allele frequencies. This function also imputes missing
# values by using the mean over all samples
wsaf_impute <- MIPanalyzer::get_wsaf(MIP_data_noDRC, impute = TRUE, FUN = mean)

# perform PCA analysis on within-sample allele frequencies
pca <- MIPanalyzer::pca_wsaf(wsaf_impute)
```

Mettons nos deux premiers composants principaux dans une base de données et produisons un nuage de points en utilisant `ggplot`:

```
# get PC1 and PC2 into dataframe
plot_df <- data.frame(PC1 = pca$x[,1],
                      PC2 = pca$x[,2],
                      Country = MIP_data_noDRC$samples$Country)

# produce scatterplot, coloured by country
ggplot(plot_df) + theme_bw() +
  geom_point(aes(x = PC1, y = PC2, color = Country))
```



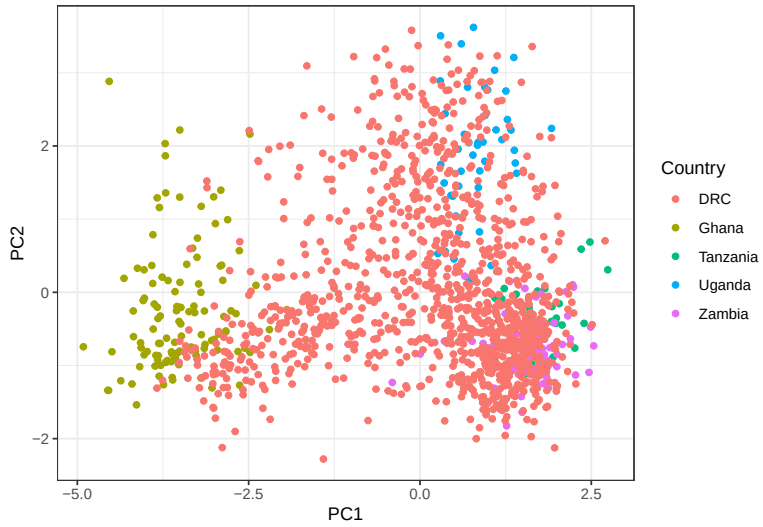
Nous pouvons voir plusieurs clusters bien séparés, correspondant à différentes zones au sein de l'Afrique subsaharienne.

Q15. Répétez le processus ci-dessus en utilisant tous les échantillons, y compris ceux de la RDC. Que remarquez-vous sur les échantillons de la RDC dans le nuage de points par rapport aux autres grappes ?

Click For Answer

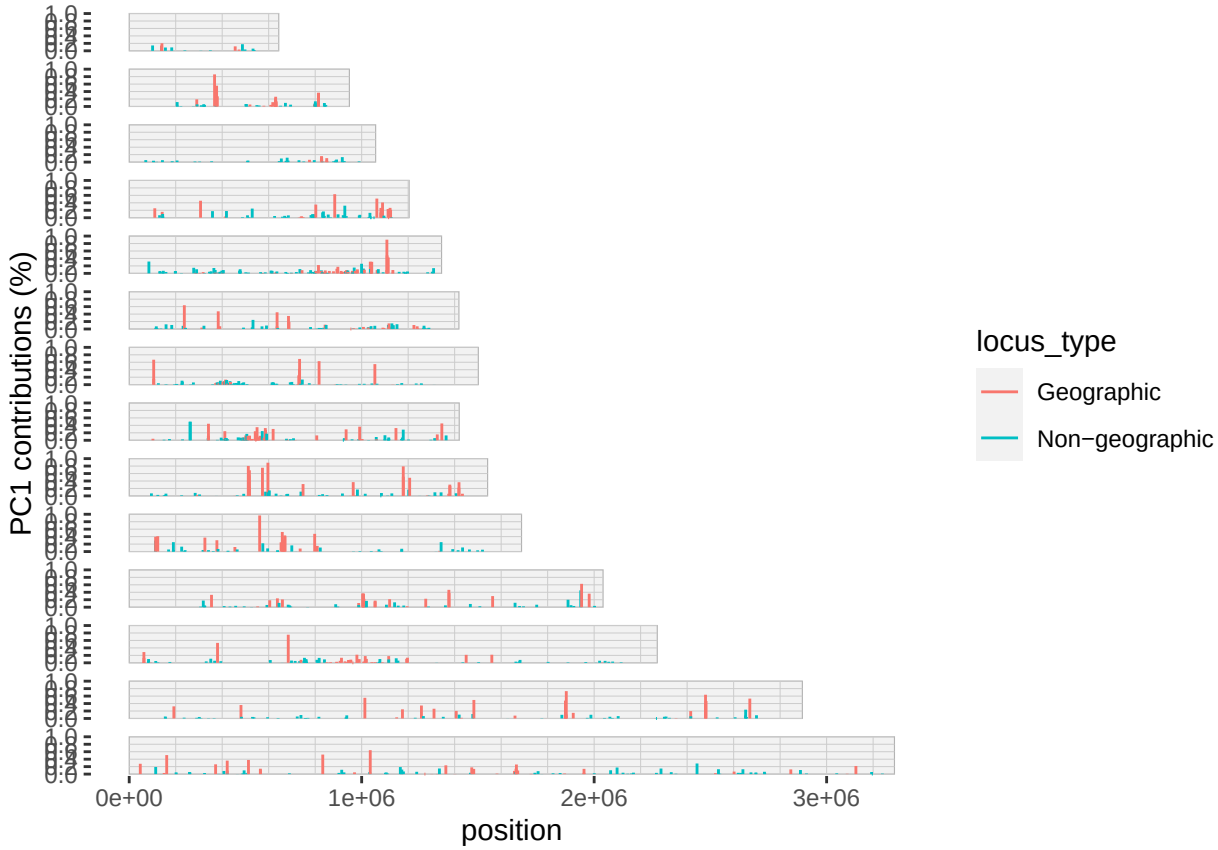
A15. Les échantillons de la RDC forment un grand groupe qui relie les trois autres. Encore une fois, cela a du sens géographiquement, car la RDC se situe entre les autres pays et nous pourrions donc nous attendre à ce que les fréquences alléliques soient intermédiaires entre les autres groupes.

```
# calculate within-sample allele frequencies
wsaf_impute <- MIPanalyzer::get_wsaf(MIP_data, impute = TRUE, FUN = mean)

# perform PCA analysis on within-sample allele frequencies
pca <- MIPanalyzer::pca_wsaf(wsaf_impute)

# get PC1 and PC2 into dataframe
plot_df <- data.frame(PC1 = pca$x[,1],
                      PC2 = pca$x[,2],
                      Country = MIP_data$samples$Country)

# produce scatterplot, coloured by country
ggplot(plot_df) + theme_bw() +
  geom_point(aes(x = PC1, y = PC2, color = Country))
```



Un autre avantage de l'ACP est que nous pouvons examiner la contribution de chaque locus à chaque composant. Ces valeurs, appelées *valeurs de chargement*, peuvent être utilisées pour déterminer quels locus sont à l'origine du modèle observé. Heureusement, la fonction `MIPanalyzer` `plot_pca_contribution()` s'en charge pour nous:

```
# plot component 1 loading values
MIPanalyzer::plot_pca_contribution(pca, component = 1, chrom = MIP_data$loci$CHROM, pos = MIP_data$loci$
                                   locus_type = MIP_data$loci$GEO)
```



Q16. Que remarquez-vous au sujet des valeurs de chargement des lieux "géographiques" par rapport aux lieux "non géographiques" ? Qu'est-ce que cela vous dit sur ce qui motive le modèle dans PC1 ?

Click For Answer

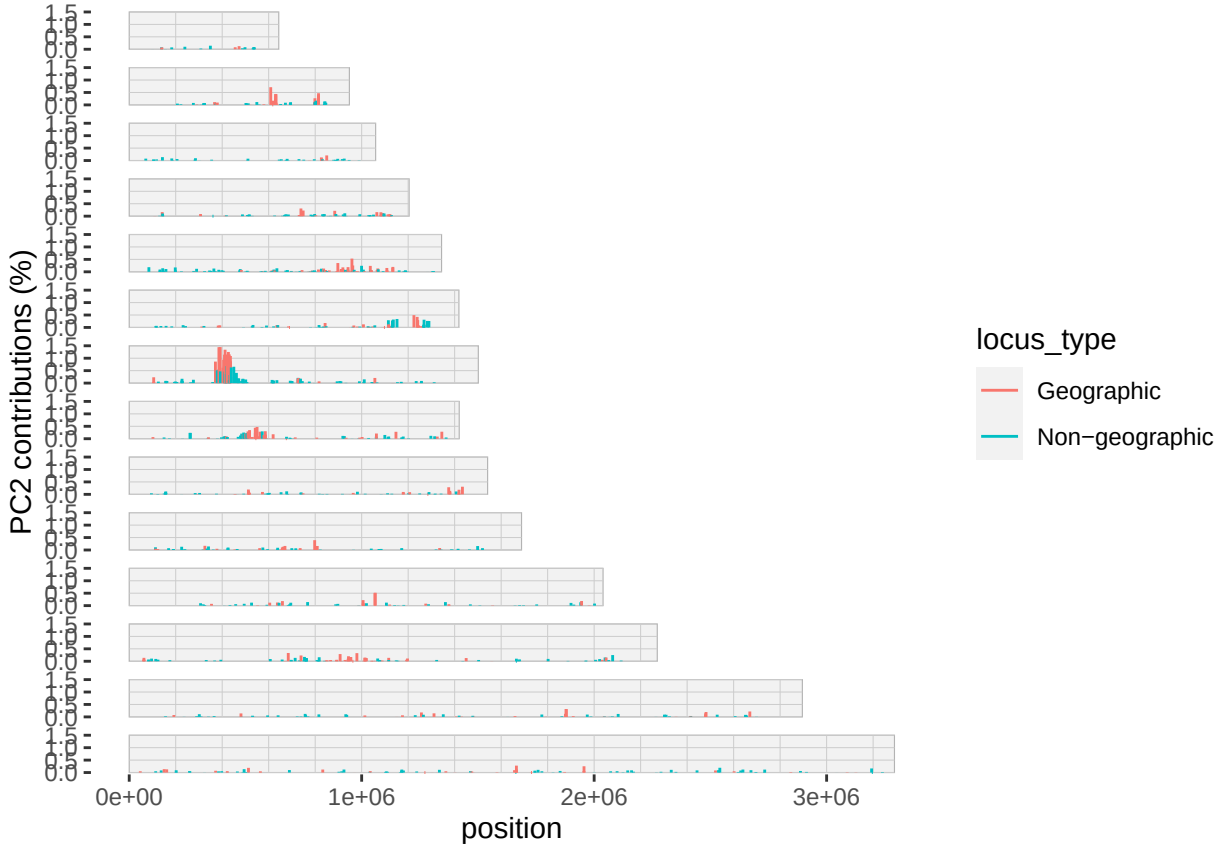
A16. Les valeurs de charge dans les lieux "géographiques" sont en moyenne plus élevées que dans les lieux "non géographiques". Ces lieux géographiques ont été choisis sur la base d'une forte différenciation spatiale sur le continent africain. Par conséquent, cela suggère que PC1 est principalement piloté par l'espace. En d'autres termes, le principal facteur expliquant la variation des fréquences alléliques que nous observons dans les données est l'endroit d'où ils ont été échantillonnés.

Q17. Répétez le processus ci-dessus, mais en regardant maintenant PC2. Que remarquez-vous à propos de l'emplacement de ces loci sur le génome ? Quels gènes sont à ces positions ? Vous pouvez utiliser PlasmoDB pour essayer d'étudier ces emplacements génomiques. Qu'est-ce que cela vous dit sur ce qui motive le modèle dans PC2 ?

Click For Answer

A17. Contrairement au PC1, nous observons désormais des pics importants à quelques endroits seulement. Il s'agit du gène *pfcrt* sur chrom7, puis d'un pic plus petit à *dhps* sur le chromosome 8. Cela nous indique que le deuxième axe de variation des fréquences alléliques est principalement lié à la résistance aux médicaments.

```
# plot component 2 loading values
MIPanalyzer::plot_pca_contribution(pca, component = 2, chrom = MIP_data$loci$CHROM, pos = MIP_data$loci$
                                   locus_type = MIP_data$loci$GEO)
```



En résumé, l'ACP peut être un outil exploratoire puissant pour trouver des modèles dans des ensembles de données de grande dimension. Lorsqu'il est appliqué aux fréquences alléliques, il peut être utilisé pour identifier les grappes (une forme de structure de population), et les valeurs de charge peuvent nous indiquer quels loci contribuent le plus fortement à ce modèle.

Déduire la structure de la population en utilisant *rmaverick*

rmaverick est un package R qui utilise une approche basée sur un modèle pour étudier la structure de la population. Elle est considérablement plus avancée que la simple approche PCA ci-dessus, qui présente certains avantages et certains inconvénients. Cette section des travaux pratiques est donc facultative, et s'adresse vraiment à ceux qui veulent découvrir des méthodes de modélisation plus avancées. Veuillez lire la documentation du package si vous souhaitez mieux comprendre les différents paramètres d'entrée et les hypothèses du modèle.

Tout d'abord, nous allons charger quelque 24 données de codes-barres génétiques SNP de Bei et al., 2018. Cette étude contient des échantillons prélevés à deux moments de plus de dix ans d'intervalle dans deux endroits au Sénégal. Nous utilisons une version traitée des données brutes qui a déjà été filtrée pour supprimer certains échantillons, et ordonnée en termes de date (2001-2002 vs 2014) et de localisation (Dielmo et Ndiop).

Les données comprennent des mesures de l'intensité de la transmission ainsi que les données brutes des codes-barres. Il comporte trois éléments : - Taux d'Inoculation Entomologique (EIR) - Les codes-barres - Les emplacements SNP

```
# load the processed data
load("data/Bei_2018_processed.RData")

names(Bei_2018_processed)
```

```
## [1] "EIR"           "barcodes"      "SNP_locations"
```

```
# take a peek at the various elements
head(Bei_2018_processed$EIR)
```

```
##   Time range Location   EIR
## 1       2001   Dielmo 353.8
## 2       2002   Dielmo 409.9
## 3       2001    Ndiop 171.9
## 4       2002    Ndiop  16.9
## 5       2014   Dielmo  26.3
## 6       2014    Ndiop <0.05
```

```
head(Bei_2018_processed$barcodes)
```

```
##   Sample Code       Date Sex  Age Parasite Density Location M/P genomic A1 B1
## 1        IP09 2001-01-02   F  3.4             9150   Dielmo           M  2  1
## 2        IP07 2001-01-03   F  2.6            20800   Dielmo           P  3  1
## 3        IP03 2001-02-05   M 10.5             6450   Dielmo           P NA  1
## 4        IP05 2001-02-05   F  3.8            22200   Dielmo           P  2  1
## 5        IP01 2001-02-06   M  2.3            10550   Dielmo           P NA NA
## 6        IP08 2002-01-07   F  3.7            29450   Dielmo           M  3  1
##   A2 B2 A3 B3 A4 B4 A5 B5 A6 B6 A7 B7 A8 B8 A9 B9 A10 B10 A11 B11 A12 B12
## 1  2  2  4  2  1  4  3  3  3  1  3  2  1  1  2  3   2   4   1   3   3   4
## 2  2  3  2  2  4  4  3 NA  3  1  2  2  2  2  2  1   2   1   2  NA   3   4
## 3 NA  3  2 NA NA  4 NA  3  3  1  2 NA  2 NA NA NA   1  NA   1   3   3   4
## 4  3  2  4  2  1  4  3  2  3 NA  3  1  2  2  2 NA   1   1   1  NA   3   4
## 5  2 NA  4  2 NA  4 NA  2  3  1 NA NA  2  2 NA NA   1  NA   1  NA   3   4
## 6  2  2  4  2  4  1  1  3  3  1  3  1  2  2  2  3   1   4   1   2   3   4
```

```
head(Bei_2018_processed$SNP_locations)
```

```
##   SNP code    SNP location
## 1       A1 Pf_01_000130573
## 2       B1 Pf_01_000539044
```

```
## 3       A2 Pf_02_000842803
## 4       B2 Pf_04_000282592
## 5       A3 Pf_05_000931601
## 6       B3 Pf_06_000145472
```

Malheureusement, `rmaverick` ne fonctionne que sur des échantillons monoclonaux. Par conséquent, filtrons tous les échantillons identifiés comme susceptibles d'être polygénomiques:

```
# filter out polygenomics
mav_data <- Bei_2018_processed$barcodes %>%
  filter(`M/P genomic` == "M")
```

Ensuite, nous devons charger les données dans *rmaverick* via la fonction `bind_data()`. Nous pouvons également configurer notre premier modèle à l'aide de la fonction `new_set()` - cela définit les hypothèses du modèle que nous allons ajuster. Nous pouvons ensuite imprimer le projet pour vérifier que tout se présente comme prévu (25 échantillons, 24 lieux, etc.)

```
# create project, bind data and setup first model
myproj <- rmaverick::mavproject() %>%
  rmaverick::bind_data(df = mav_data, ID_col = 1, data_cols = 8:31, ploidy = 1) %>%
  rmaverick::new_set(name = "no admixture model", admix_on = FALSE)

myproj
```

```
## DATA:
##    individuals = 25
##    loci = 24
##    ploidy = 1
##    pops = 1
##    missing data = 2 of 600 gene copies (0%)
##
## PARAMETER SETS:
##  * SET1: no admixture model
##
## ACTIVE SET: SET1
##    model = no-admixture
##    lambda = 1
```

Nous sommes maintenant prêts à exécuter l'analyse principale. Cela utilise Markov Chain Monte Carlo (MCMC) pour échantillonner à partir de la distribution a posteriori de l'appartenance au groupe. En d'autres termes, il essaie de regrouper des échantillons qui ont des allèles similaires sur plusieurs locus. Une question importante est - combien de groupes (sous-populations) y a-t-il ? Le nombre de groupes est nommé K dans ce modèle, et l'analyse consiste à explorer plusieurs valeurs différentes de K et à déterminer celle qui, selon nous, est la mieux étayée par les données.

La fonction suivante exécute l'analyse principale `rmaverick`. Vous devriez l'exécuter sans le wrapper `quite()` afin que vous puissiez voir le MCMC complet en action. Notez que nous explorons ici les valeurs de K de 1 à 8 :

```
# run main analysis
myproj <- quiet(rmaverick::run_mcmc(myproj, K = 1:8, burnin = 1e3, samples = 1e3,
                                    rungs = 10, GTI_pow = 1.5))
```

Il existe divers diagnostics que nous devons utiliser pour vérifier que notre MCMC a fonctionné comme prévu. Pour gagner du temps, nous les passerons ici, mais veuillez lire la [documentation du package](https://github.com/bobverity/rmaverick) lorsque vous exécutez sur vos propres données pour vous assurer d'obtenir des résultats significatifs.

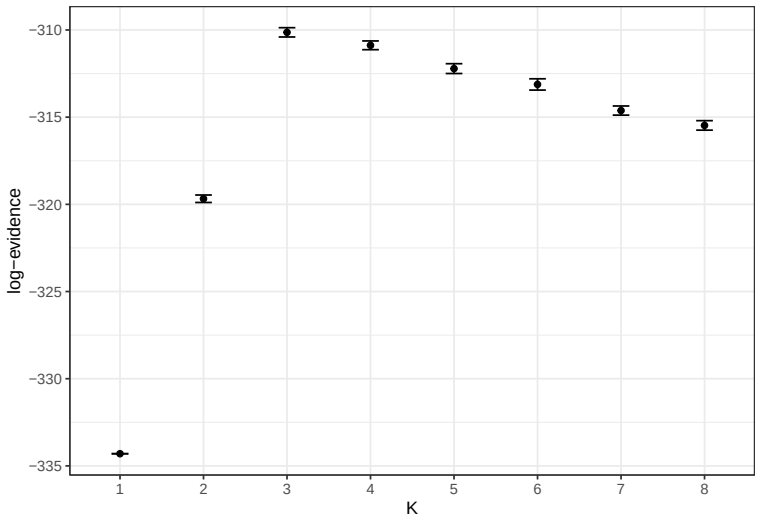
La fonction `plot_logevidence_K()` nous montre la log-vraisemblance de chaque valeur de K explorée. Les valeurs positives (ou les valeurs moins négatives) offrent une meilleure prise en charge de K. La fonction `plot_posterior_K()` fonctionne de manière très similaire, mais peut être interprétée comme une probabilité ordinaire.

Q18. Utilisez les fonctions `plot_logevidence_K()` et `plot_posterior_K()` pour déterminer quelle valeur de K est la mieux prise en charge dans ce cas. Quelles valeurs sont prises en charge ?

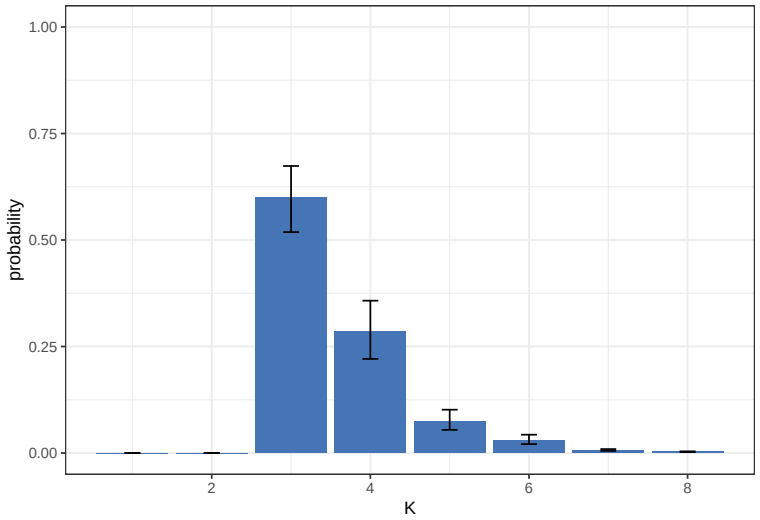
Click For Answer

A18. D'après les deux tracés, il est clair que $K = 3$ est le mieux pris en charge dans ce cas. À partir du diagramme de probabilité, nous pouvons voir que les valeurs de $K = 4$ et $K = 5$ sont également plausibles.

```
# plot in log space
rmaverick::plot_logevidence_K(myproj)
```




```
# plot as probability
rmaverick::plot_posterior_K(myproj)
```



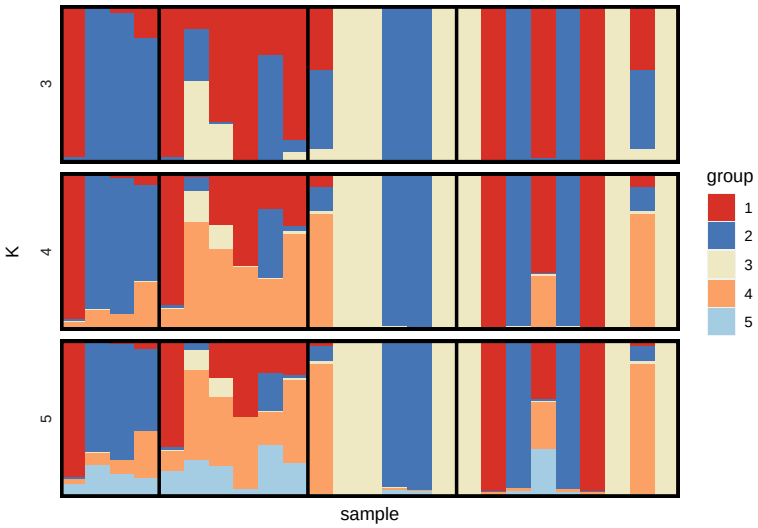
Enfin, nous pouvons utiliser la fonction `plot_qmatrix()` pour produire un tracé "STRUCTURE", nommé d'après le programme STRUCTURE original sur lequel `rmaverick` est basé. Ce tracé peut être interprété de la manière suivante :

- Chaque barre empilée représente un échantillon.

- La proportion de chaque couleur représente la probabilité que cet échantillon appartienne à chaque grappe. Par exemple, si une barre est à 50 % rouge et à 50 % bleue, cet échantillon a une chance égale d'appartenir à chacun de ces deux groupes.
- Nous produisons souvent plusieurs tracés STRUCTURE à la fois pour différentes valeurs de K. Le nombre de couleurs présentes dans le tracé sera égal à K.

Dans ce cas, nous ajoutons également des lignes verticales pour répartir les données génétiques dans les groupes suivants : 1. Dielmo 2001-2001 2. Ndiop 2001-2002 3. Dielmo 2014 4. Ndiop 2014

```
rmaverick::plot_qmatrix(myproj, K = 3:5) +
  geom_vline(xintercept = c(4, 10, 16) + 0.5, size = 1)
```



Notez que les derniers échantillons sont (pour la plupart) clairement attribués à l'une ou l'autre population avec une probabilité élevée. Ceux-ci représentent essentiellement des lignées clonales. Nous pouvons voir que les groupes 2,3,4 sont présents à Dielmo en 2014, et à Ndiop nous avons en plus le groupe 1.

En comparant cela au point temporel précédent, les échantillons sont beaucoup plus ambigus dans leur allocation. Ils ressemblent un peu à certaines de ces lignées clonales, mais ce n'est pas clair. Fait intéressant, le groupe 4 domine à Ndiop en 2001-2002, alors qu'il était presque totalement absent de Ndiop en 2014.

Q19. Dans l'ensemble, nous pouvons constater une augmentation de la structure de la population au fil du temps. Examinez maintenant les valeurs EIR, également stockées dans le même ensemble de données. Pouvez-vous donner un sens à ce qui se passe probablement ici ?

Click For Answer

A19. Il y a une transmission plus faible au dernier moment, il n'est donc pas surprenant que la population devienne plus clonale. Nous constatons également une différenciation croissante au fil du temps, comme on pourrait s'y attendre si les populations étaient partiellement isolées. Cependant, nous devons garder à l'esprit qu'il ne s'agit que d'un petit échantillon à chaque instant et qu'il est donc probable que de nombreux génotypes de la population aient été omis.